AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior version, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) In a J2EE application server, a method for monitoring performance of a plurality of transactions including a top level transaction and plurality of transactions relating to said top level transaction in a child parent hierarchy, comprising

for each of selected ones of said plurality of transactions, obtaining a performance metric corresponding to the selected transaction by:

installing an instrument hook upon loading the selected transaction; and

instrumenting said selected transaction upon execution of the selected transaction using one or more plug-in instruments called by the instrument hook;

~~instrumenting said transaction at run time without modifying its source code to obtain a performance metric corresponding thereto,~~

for each of said instrumented transactions, generating a correlator for identifying said top level transaction and a parent transaction, if any, of said instrumented transaction, and

utilizing said correlators to cross-correlate a performance metric corresponding to a parent transaction with one or more performance metrics corresponding to one or more child transactions of said parent transaction.

2. (Currently Amended) The method of claim 1, wherein the step of instrumenting said selected transaction comprises inserting instrumentation code in a bytecode representation of said selected transaction.

3. (Currently Amended) The method of claim 2, wherein said performance metric corresponds to a response time of said selected transaction.

4. (Currently Amended) The method of claim 3, wherein said instrumentation code effects generation of a start time marker upon start of execution of said selected transaction and generation of a stop time marker upon completion of execution of said selected transaction.

5. (Original)   The method of claim 4, wherein said instrumentation code generates calls to an Application Response Measurement (ARM) agent to cause generation of said stop and start time markers.

6. (Currently Amended) The method of claim 5, further comprising utilizing said start and stop time markers to measure a response time of said <u>selected</u> transaction.

7. (Currently Amended) The method of claim 1, further comprising generating a record for each instrumented transaction upon completion of said <u>instrumented</u> transaction, said record indicating said performance metric associated with said <u>instrumented</u> transaction, a parent of said <u>instrumented</u> transaction, and said top level transaction.

8. (Currently Amended) The method of claim 7, further comprising transmitting said <u>instrumented</u> transaction record to an analysis and presentation module.

9. (Original) The method of claim 1, further comprising storing said correlators in a thread local storage stack in case of execution of said hierarchical transactions in a single thread.

10. (Original) The method of claim 9, further comprising storing said correlators in the stack based on a LIFO protocol.

11. (Currently Amended) The method of claim 10, further comprising removing <u>one</u> a correlator<u> of said correlators</u> from said stack upon completion of <u>said hierarchical</u> a transaction associated with said correlator.

12. (Original) The method of claim 1, wherein said top level transaction is initiated in response to a request received from a web server.

13. (Original) The method of claim 12, wherein said web server transmits a cookie to said application server together with said request.

14. (Original) The method of claim 13, further comprising

utilizing said cookie to generate said top level correlator.

15. (Currently Amended) A method for monitoring performance of at least two Java transactions that are ~~executing in two separate processes and being~~ related to one another as parent-child transactions, comprising

obtaining a performance metric corresponding to each of said at least two Java transactions by:

installing an instrument hook upon loading each of said at least two Java transactions; and

instrumenting each of said at least two Java transactions upon execution of each of said at least two Java transactions using one or more plug-in instruments called by the instrument hook;

~~instrumenting each of said transactions at run time by modifying its respective bytecode representation to obtain a selected performance metric corresponding thereto,~~

generating a correlator corresponding to said parent ~~top-level~~ transaction,

utilizing RMI over IIOP to send said parent ~~top-level~~ correlator incorporated in a header of an IIOP message to said child transaction, and

generating another correlator corresponding to said child transaction.

16. (Currently Amended) The method of claim 15, further comprising

employing said correlators to cross correlate the performance metric of said top parent ~~top-level~~ transaction with the performance metric of said child transaction.

17. (Currently Amended) The method of claim 15, wherein said performance metric corresponds to a response time of each of said at least two Java transactions ~~said transaction~~.

18. (Currently Amended) The method of claim 17, wherein said performance metric corresponds to ~~modified bytecode representation effects generation of~~ a start time marker upon start of execution of each of said at least two Java transactions ~~said transaction~~ and a stop time marker upon completion of execution of each of said at least two Java transactions ~~said transaction~~.

19. (Original) The method of claim 18, wherein said modified bytecode representation generate calls to an Application Response Measurement (ARM) agent to cause generation of said start and stop time markers.

20. (Currently Amended) A computer readable medium instructions operable by a computer which when executed cause the computer to perform a method comprising:

obtaining a performance metric corresponding to selected transaction of a plurality of parent-child transactions by

installing an instrument hook upon loading the selected transaction; and

instrumenting said selected transaction upon execution of the selected transaction using one or more plug-in instruments called by the instrument hook; and

~~instrumenting at run-time a hierarchical chain of parent-child transactions including a top-level transaction and at least one child transaction thereof without modifying source codes associated with these transactions; and~~

generating correlators for each of said transactions, wherein each correlator identifies said top level transaction and a parent transaction, if any, corresponding to its associated transaction.